ZILKA·KOTAB
———— PC ————
ZILKA, KOTAB & PEECE™

100 PARK CENTER PLAZA, SUITE 300
SAN JOSE, CA 95113

TELEPHONE (408) 971-2573
FAX (408) 971-4660

FAX COVER SHEET

Date:	November 27, 2006	Phone Number	Fax Number
To:	Board of Patent Appeals		(571) 273-8300
From:	Kevin J. Zilka		

Docket No.:   NAI1P451/01.285.01   **App. No: 10/092,420**

Total Number of Pages Being Transmitted, Including Cover Sheet: 40

Message:

Please deliver to the Board of Patent Appeals.

Thank you,

Kevin J. Zilka

□ *Original to follow Via Regular Mail* **X** *Original will Not be Sent*   □ *Original will follow Via Overnight Courier*

**IF YOU DO NOT RECEIVE ALL PAGES OR IF YOU ENCOUNTER
ANY OTHER DIFFICULTY, PLEASE PHONE _____Erica_____
AT (408) 971-2573 AT YOUR EARLIEST CONVENIENCE**

November 27, 2006

RECEIVED
CENTRAL FAX CENTER

NOV 2 7 2006

Practitioner's Docket No. NAI1P451/01.285.01   *PATENT*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Manuel Nedbal et al.

Application No.: 10/092,420	Group No.: 2142
Filed: 03/07/2002	Examiner: Lin, K.

For: AGENT ARCHITECTURE FOR TRIGGERING REMOTELY INITIATED DATA PROCESSING
OPERATIONS

Mail Stop Appeal Briefs – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF
(PATENT APPLICATION--37 C.F.R. § 41.37)

1.   This brief is in furtherance of the Notice of Appeal, filed in this case on 06/12/2006 and in response
to the Notice of Panel Decision from Pre-Appeal Brief Review mailed 07/28/2006.

2.   STATUS OF APPLICANT

This application is on behalf of other than a small entity.

CERTIFICATION UNDER 37 C.F.R. §§ 1.8(a) and 1.10*
(When using Express Mail, the Express Mail label number is mandatory;
Express Mail certification is optional.)

I hereby certify that, on the date shown below, this correspondence is being:

MAILING
_ deposited with the United States Postal Service in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA
22313-1450.

37 C.F.R. § 1.8(a)	37 C.F.R. § 1.10*
_ with sufficient postage as first class mail.	_ as "Express Mail Post Office to Addressee"
	Mailing Label No. _____ (mandatory)

TRANSMISSION
✓ facsimile transmitted to the Patent and Trademark Office, (571) 273 - 8300.

Date: 11/27/2006

Signature

Erica L. Farlow

(type or print name of person certifying)

* Only the date of filing (' 1.6) will be the date used in a patent term adjustment calculation, although the date on any certificate of mailing or
transmission under ' 1.8 continues to be taken into account in determining timeliness. See ' 1.703(f). Consider "Express Mail Post Office to
Addressee" (' 1.10) or facsimile transmission (' 1.6(d)) for the reply to be accorded the earliest possible filing date for patent term adjustment
calculations.

Transmittal of Appeal Brief--page 1 of 2

3. FEE FOR FILING APPEAL BRIEF

Pursuant to 37 C.F.R. § 41.20(b)(2), the fee for filing the Appeal Brief is:

other than a small entity	$500.00
Appeal Brief fee due	**$500.00**

4. EXTENSION OF TERM

The proceedings herein are for a patent application and the provisions of 37 C.F.R. § 1.136 apply.

Applicant petitions for an extension of time under 37 C.F.R. § 1.136 (fees: 37 C.F.R. § 1.17(a)(1)-(5)) for three months:

Fee: $1,020.00

If an additional extension of time is required, please consider this a petition therefor.

Applicant believes that no extension of term is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

5. TOTAL FEE DUE

The total fee due is:

Appeal brief fee	$500.00
Extension fee (if any)	$1,020.00
TOTAL FEE DUE	**$1,520.00**

6. FEE PAYMENT

Authorization is hereby made to charge the amount of $1,520.00 to Deposit Account No. 50-1351 (Order No. NAI1P451).

A duplicate of this transmittal is attached.

7. FEE DEFICIENCY

If any additional extension and/or fee is required, and if any additional fee for claims is required, charge Deposit Account No. 50-1351 (Order No. NAI1P451).

Reg. No.: 41,429
Tel. No.: 408-971-2573
Customer No.: 28875

Signature of Practitioner
Kevin J. Zilka
Zilka-Kotab, PC
P.O. Box 721120
San Jose, CA 95172-1120
USA

Transmittal of Appeal Brief—page 2 of 2

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:)	
)	
Nedbal et al.)	Group Art Unit: 2142
)	
Application No. 10/092,420)	Examiner: Lin, Kelvin Y.
)	
Filed: March 7, 2002)	Date: November 27, 2006
)	
For: AGENT ARCHITECTURE FOR)	
TRIGGERING REMOTELY INITIATED)	
DATA PROCESSING OPERATIONS)	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

ATTENTION: Board of Patent Appeals and Interferences

APPEAL BRIEF (37 C.F.R. § 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on 06/12/2006 and in response to the Notice of Panel Decision from Pre-Appeal Brief Review mailed 07/28/2006.

The fees required under § 1.17, and any required petition for extension of time for filing this brief and fees therefor, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief contains these items under the following headings, and in the order set forth below (37 C.F.R. § 41.37(c)(i)):

I	REAL PARTY IN INTEREST
II	RELATED APPEALS AND INTERFERENCES
III	STATUS OF CLAIMS
IV	STATUS OF AMENDMENTS
V	SUMMARY OF CLAIMED SUBJECT MATTER
VI	GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

- 2 -

The final page of this brief bears the practitioner's signature.

- 3 -

I REAL PARTY IN INTEREST (37 C.F.R. § 41.37(c)(1)(i))

The real party in interest in this appeal is McAfee, Inc.

- 4 -

II RELATED APPEALS AND INTERFERENCES (37 C.F.R. § 41.37(c) (1)(ii))

With respect to other prior or pending appeals, interferences, or related judicial proceedings that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no other such appeals, interferences, or related judicial proceedings.

- 5 -

III STATUS OF CLAIMS (37 C.F.R. § 41.37(c) (1)(iii))

A.   TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1-4, 9, 14-21, 26, 31-38, 43, and 48-54

B.   STATUS OF ALL THE CLAIMS IN APPLICATION

1.   Claims withdrawn from consideration: None
2.   Claims pending: 1-4, 9, 14-21, 26, 31-38, 43, and 48-54
3.   Claims allowed: None
4.   Claims rejected: 1-4, 9, 14-21, 26, 31-38, 43, and 48-54
5.   Claims cancelled: 5-8, 10-13, 22-25, 27-30, 39-42, and 44-47

C.   CLAIMS ON APPEAL

The claims on appeal are: 1-4, 9, 14-21, 26, 31-38, 43, and 48-54

See additional status information in the Appendix of Claims.

- 6 -

IV STATUS OF AMENDMENTS (37 C.F.R. § 41.37(c)(1)(iv))

As to the status of any amendment filed subsequent to final rejection, amendments were made in Amendment B mailed February 15, 2006, and such amendment were entered, as confirmed in the Advisory Action mailed April 10, 2006.

V SUMMARY OF CLAIMED SUBJECT MATTER (37 C.F.R. § 41.37(c)(1)(v))

With respect to a summary of Claim 1, as shown in Figures 1-28, a computer program product for controlling a target computer to perform an operation in response to data received from an initiating computer is provided. In use, agent process code is operable to execute on the target computer to provide an agent process (e.g. see item 300 of Figure 1, etc.). The agent process executing on the target computer receives autonomously generated operation specifying data sent from the initiating computer to the target computer (e.g. see item 336 of Figure 5 and item 344 of Figure 6, etc.). Further, an identifier of a target process for performing said operation is read from the operation specifying data (e.g. see item 348 of Figure 6, etc.). Also, if said target process is available to said target computer, at least a portion of said operation specifying data is passed from said agent process to said target process (e.g. see item 350 of Figure 6, etc.).

In addition, target process code is operable to provide one or more target processes for performing operations in response to operation specifying data (e.g. see item 362 of Figure 7, etc.). Further, the one or more target processes are provided at the target computer independently of the agent process. The operation performed includes configuring the target computer to execute a computer program (e.g. see item 362 of Figure 7, etc.). In addition, the target process is operable to map configuration data specified within said operation specifying data to a configuration data store of said target computer (e.g. see Figure 3, etc.). The configuration data store is one of a Windows Registry entry; an INI file; a DAPI store; and a database entry (e.g. see Figures 12, 13, and 14, etc.). Also, the identifier of the target process includes at least one of data specifying a computer file operable to trigger the target process, data specifying a communication channel operable to trigger the target process; and data specifying an operating system command operable to trigger the target process.

The operation includes returning result data from the target computer to the initiating computer in dependence upon the operation performed by the target process (e.g. see item 364 of Figure 7, etc.). Further, the result data includes data specifying existing configuration data of said target computer. In addition, the target process is operable to map the existing configuration data of the target computer stored within the configuration data store of the target computer to the result

- 8 -

data to be returned to the initiating computer (see Figure 4, etc.). See, for example, page 9, lines 33-34; page 10, lines 6-24; page 11, lines 6-33; page 12, lines 6-23; and page 12, lines 25-34 et al.

With respect to a summary of Claim 18, as shown in Figures 1-28, a method of controlling a target computer to perform an operation in response to data received from an initiating computer is provided. In use, autonomously generated operation specifying data sent from the initiating computer to the target computer is received at an agent process (e.g. see item 300 of Figure 1, etc.) executing on the target computer (e.g. see item 336 of Figure 5 and item 344 of Figure 6, etc.). Further, an identifier of a target process for performing the operation is read from the operation specifying data (e.g. see item 348 of Figure 6, etc.). In addition, at least a portion of said operation specifying data is passed from said agent process to said target process if the target process is available to the target computer (e.g. see item 350 of Figure 6, etc.).

Still yet, the operation is performed using the target process (e.g. see item 362 of Figure 7, etc.). One or more target processes for performing operations in response to operation specifying data (e.g. see Figure 7, etc.) are provided at the target computer independently of the agent process (e.g. see item 362 of Figure 7, etc.). The operation performed includes configuring the target computer to execute a computer program (e.g. see item 362 of Figure 7, etc.). Moreover, the target process is operable to map configuration data specified within said operation specifying data to a configuration data store of said target computer (e.g. see Figure 3, etc.). Further, the configuration data store is one of a Windows Registry entry; an INI file; a DAPI store; and a database entry (e.g. see Figures 12, 13, and 14, etc.). Also, the identifier of a target process includes at least one of data specifying a computer file operable to trigger the target process, data specifying a communication channel operable to trigger the target process, and data specifying an operating system command operable to trigger the target process.

Additionally, the operation includes returning result data from the target computer to the initiating computer in dependence upon the operation performed by the target process (e.g. see item 364 of Figure 7, etc.). Further, the result data includes data specifying existing configuration data of said target computer. Also, the target process is operable to map the existing configuration data of the target computer stored within the configuration data store of

- 9 -

the target computer to the result data to be returned to the initiating computer. See, for example, page 9, lines 33-34; page 10, lines 6-24; page 11, lines 6-33; page 12, lines 6-23; and page 12, lines 25-34 et al.

With respect to a summary of Claim 35, as shown in Figures 1-28, an apparatus is provided for controlling a target computer to perform an operation in response to data received from an initiating computer. In use, agent process logic of the target computer provides an agent process (e.g. see item 300 of Figure 1, etc.) to receive, at an agent process executing on the target computer, autonomously generated operation specifying data sent from the initiating computer to the target computer (e.g. see item 336 of Figure 5 and item 344 of Figure 6, etc.). Further, an identifier of a target process for performing said operation is read from the operation specifying data (e.g. see item 348 of Figure 6, etc.). In addition, at least a portion of said operation specifying data is passed from said agent process to said target process if said target process is available to said target computer (e.g. see item 350 of Figure 6, etc.).

Target process logic is operable to provide one or more target processes for performing operations in response to operation specifying data (e.g. see item 362 of Figure 7, etc.). The one or more target processes are provided at the target computer independently of the agent process. Further, the operation performed includes configuring the target computer to execute a computer program (e.g. see item 362 of Figure 7, etc.). The target process is operable to map configuration data specified within said operation specifying data to a configuration data store of said target computer (e.g. see Figure 3, etc.). The configuration data store is one of a Windows Registry entry; an INI file; a DAPI store; and a database entry (e.g. see Figures 12, 13, and 14, etc.). Also, the identifier of the target process includes at least one of data specifying a computer file operable to trigger the target process, data specifying a communication channel operable to trigger the target process, and data specifying an operating system command operable to trigger the target process.

In addition, the operation includes returning result data from the target computer to the initiating computer in dependence upon the operation performed by the target process (e.g. see item 364 of Figure 7, etc.). The result data includes data specifying existing configuration data of said target computer. The target process is operable to map the existing configuration data of the target

- 10 -

computer stored within the configuration data store of the target computer to the result data to be returned to the initiating computer. See, for example, page 9, lines 33-34; page 10, lines 6-24; page 11, lines 6-33; page 12, lines 6-23; and page 12, lines 25-34 et al.

- 11 -

**VI GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL (37 C.F.R. §
41.37(c)(1)(vi))**

Following, under each issue listed, is a concise statement setting forth the corresponding ground
of rejection.

Issue # 1: The Examiner has rejected Claims 1-4, 9, 14-21, 26, 31-38, 43 and 48-54 under 35 U.S.C.
103(a) as being unpatentable over Uszok et al. (U.S. Patent Application No. 2004/0205772) in view
of Kouznetsov et al. (U.S. Patent No. 6,931,546).

- 12 -

VII ARGUMENT (37 C.F.R. § 41.37(c)(1)(vii))

The claims of the groups noted below do not stand or fall together. In the present section, appellant explains why the claims of each group are believed to be separately patentable.

<u>Issue # 1:</u>

The Examiner has rejected Claims 1-4, 9, 14-21, 26, 31-38, 43 and 48-54 under 35 U.S.C. 103(a) as being unpatentable over Uszok et al. (U.S. Patent Application No. 2004/0205772) in view of Kouznetsov et al. (U.S. Patent No. 6,931,546).

>   *Group #1: Claims 1, 16, 18, 33, 35, 50, and 53*

With respect to independent Claims 1, 18, and 35, the Examiner has relied on paragraph [0079] lines 1-15 and paragraph [0080] lines 1-8 in Uszok to make a prior art showing of appellant's claimed "if said target process is available to said target computer to pass at least a portion of said operation specifying data from said agent process to said target process" (see the same or similar, but not necessarily identical language in each of the independent claims).

In addition, in the Office Action dated 12/23/2005, the Examiner has argued that Figure 4, paragraph [0079] lines 1-15, and paragraph [0080] lines 1-8 in Uszok teach that "the plug-in is available to the target computer (botServer) and the plug-ins can be configured so that they have the same ID on every server." The Examiner has further stated that such excerpts teach that "plug-ins can be installed by botServer manager (which is the agent process) [which] communicates with other plug-ins and bots via botServer manager using messages...[such that] the plug-in can be passed from the agent process (botServer manager) to [the] target process (plug-in manager), and both agent, and target processes are performed at the same target computer (botServer)."

After careful review of the Examiner's arguments in association with the excerpts from Uszok relied on by the Examiner, appellant notes that Uszok does not teach passing "at least a portion of said operation specifying data from said agent process to said target process," as claimed by

- 13 -

appellant. In particular, appellant notes that the Examiner has relied on the botServer manager in Uszok to meet appellant's claimed "agent process," however, appellant points out that the botServer manager in Uszok only allows plug-ins to communicate with other plug-ins and bots (see specifically paragraph [0079] lines 2-5). Clearly, a botServer manager that only provides a communication channel between plug-ins and other plug-ins and bots does <u>not</u> meet appellant's claimed agent process from which "at least a portion of said <u>operation specifying data</u> [is passed]" (emphasis added), as claimed by appellant.

Furthermore, it seems the Examiner has equated Uszok's "plug-in" with appellant's claimed "operation specifying data," since the Examiner has stated that "the plug-in can be passed from the agent process...to [the] target process." Appellant respectfully points out that, in Uszok, the plug-in itself "<u>perform</u>[s] demanding real-time operations" (see paragraph [0079] lines 13-15 – emphasis added), which clearly cannot meet any sort of "operation specifying <u>data</u>" (emphasis added), as claimed by appellant.

In the Advisory Action mailed 04/10/2006, the Examiner has argued that paragraph [0092] in Uszok discloses "the revision of bot configuration or request a progress update are functioning as a portion of operation specifying data." Appellant respectfully asserts that the "heartbeat" relied upon by the Examiner is merely disclosed in a context where the 'botMaster Communication Manager sends the message to the botBox during the next synchronization "heartbeat".' However, the excerpts from Uszok relied upon by the Examiner still fail to even suggest a technique where "if said target process is available to said target computer to pass at least a portion of said operation specifying data <u>from said agent process to said target process</u>" (emphasis added), as claimed by appellant.

Still with respect to each of the independent claims, the Examiner has relied on paragraph [0079] lines 1-15, paragraph [0083] lines 1-8, and paragraph [0084] lines 1-6 in Uszok to make a prior art showing of appellant's claimed "target process code operable to provide one or more target processes for performing operations in response to operation specifying data, said one or more target processes being provided at said first computer independently of said agent process" (see the same or similar, but not identical language in each of the independent claims).

- 14 -

After reviewing the Examiner's arguments and the related excerpts in Uszok, appellant notes that the Examiner has relied on Uszok's Security Registry to meet appellant's claimed "one or more target processes." Appellant points out, however, that the Security Registry in Uszok only includes lists of data, such that it is only a storage space and not any sort of process, as claimed by appellant. It also seems that the Examiner has again relied on Uszok's plug-ins to meet appellant's claimed "operation specifying data." As noted above, Uszok's plug-ins perform operations, which clearly cannot meet any sort of operation specifying data, as claimed by appellant.

In addition, the Examiner has relied on paragraph [0144] lines 1-13 in Uszok to make a prior art showing of appellant's claimed technique "wherein said operation performed includes configuring said target computer to execute a computer program." Appellant respectfully asserts that such excerpt only teaches configuring an sBot which is located on a botServer (see Figure 4 and paragraph [0081]), and not a target computer, as claimed by appellant. In particular, appellant notes that the Examiner has previously relied on Uszok's botServer to meet appellant's claimed "target computer" in prior claim limitations including, for example, those argued above. Thus, configuring an sBot cannot meet appellant's claimed "configuring said target computer," as claimed. Furthermore, the sBot in Uszok is only configured to clone itself and/or migrate to a new server (paragraph [0144]), and is not configured "to execute a computer program," as appellant claims.

Furthermore, with respect to each of the independent claims, the Examiner has relied on Col. 9, lines 59-65 and Col. 12, lines 9-20 in Kouznetsov to make a prior art showing of appellant's claimed techniques "wherein said target process is operable to map configuration data specified within said operation specifying data to a configuration data store of said target computer" and "wherein said configuration data store is one of: a Windows Registry entry; an INI file; a DAPI store; and a database entry." The Examiner has specifically stated that Kouznetsov teaches that the initialization file "corresponds to map the configuration specified within (embedded) said operation using .ini files."

Appellant notes, however, that such excerpts only disclose a configuration file comprising an initialization file embedded within a cabinet file. Clearly, simply disclosing a configuration file

- 15 -

comprising an initialization file within a cabinet file does not even suggest <u>mapping</u>, let alone mapping "configuration data specified within said operation specifying data <u>to a configuration data store of said target computer</u>" (emphasis added), as appellant claims. In addition, Kouznetsov merely teaches that the "received messages <u>identify</u> a configuration file" (see Col. 9, line 61 – emphasis added), but not that "configuration data [is] specified within said operation specifying data," where such operation specifying data is utilized in "configur[ing] said target computer to execute a computer program," in the context claimed by appellant (see each of the independent claims).

In the Advisory Action mailed 04/10/2006, the Examiner has argued that paragraphs [0092] and [0111] in Uszok disclose a technique "wherein said target process is operable to map configuration data specified within said operation specifying data to a configuration data store of said target computer," as claimed by appellant. Appellant respectfully asserts that Uszok merely teaches that "[t]he botBox Manager forwards the message to botBox, e.g., 414, and botBox finds the appropriate sBot." To this end, the excerpts from Uszok simply fail to disclose a technique "wherein said target process is operable to <u>map configuration data</u> specified <u>within said operation specifying data</u>..." (emphasis added), as claimed by appellant. In addition, Uszok suggests that "[t]he configuration block 722 can be used for selecting or configuring a required logical function to be carried out by the logic block 702," which fails to even suggest a technique "wherein said target process is operable to map configuration data specified within said operation specifying data <u>to a configuration data store of said target computer</u>" (emphasis added), as claimed by appellant. Furthermore, there is simply no disclosure in the excerpts from Uszok relied upon by the Examiner that meet a technique "wherein said configuration data store is one of: a Windows Registry entry; an INI file; a DAPI store; and a database entry," as specifically claimed by appellant.

Still yet, with respect to each of the independent claims, the Examiner has relied on paragraph [0068] lines 3-10 in Uszok to make a prior art showing of appellant's claimed technique "wherein said identifier of a target process includes at least one of: data specifying a computer file operable to trigger said target process; data specifying a communication channel operable to trigger said target process; and data specifying an operating system command operable to trigger

said target process." Specifically, the Examiner has stated that such excerpt teaches "using the communication channel to trigger the BotBox initialization."

Appellant respectfully asserts that Uszok only teaches that "establishing a botBox is initiated by the user or prompted by botMaster through the GUI...[and that] a botBox communicator component...[is used] to send a message to request initialization of a corresponding botBox." Clearly, such teachings in Uszok do not even suggest "data specifying a communication channel" (emphasis added), as claimed by appellant, especially since Uszok only generally teaches that the botBox proxy "utilizes a...botBox communicator component." Furthermore, when read in the appropriate context, appellant's claimed "identifier of a target process" is "read from said operation specifying data." Simply nowhere does Uszok teach "data specifying a communication channel" which is read from operation specifying data, in the specific context claimed by appellant.

Further, with respect to the independent claims, the Examiner has relied on paragraph [0122] lines 1-2 in Uszok in rejecting appellant's claimed technique "wherein said result data includes data specifying existing configuration data of said target computer." Appellant respectfully asserts that such excerpt from Uszok only teaches user profiles that may be created on a temporary basis. Appellant notes that the Examiner has argued that the "profile [in Uszok] corresponds to configuration data." Appellant respectfully asserts that in Uszok, there is no "result data [that] includes data specifying existing" user profiles, when taken in the context of appellant's claim language. Thus, simply nowhere does Uszok specifically teach "result data [that] includes data specifying existing configuration data of said target computer," as claimed by appellant.

Additionally, with respect to the independent claims, the Examiner has relied on paragraph [0100] lines 1-15 in Uszok to make a prior art showing of appellant's claimed technique "wherein said target process is operable to map said existing configuration data of said target computer stored within said configuration data store of said target computer to said result data to be returned to said initiating computer." Appellant respectfully asserts that such excerpt merely teaches that "the portable program assigns to its mBots one of the user profiles made available on the shared botBox." Clearly, the user profiles in Uszok are only being used with respect to the

- 17 -

target computer to which the mBots are assigned, and are not utilized by being mapped "to said result data to be returned to said initiating computer" (emphasis added), as claimed by appellant.

To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on appellant's disclosure. In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed.Cir.1991).

Appellant respectfully asserts that at least the third element of the prima facie case of obviousness has not been met, since the prior art references, when combined, fail to teach or suggest all of the claim limitations, as noted above.

   Group #2: Claims 2, 19, and 36

With respect to the Claims 2, 19, and 36, the Examiner has relied on paragraph [0014] line 11 from Uszok to make a prior art showing of appellant's claimed technique "wherein said operation specifying data is passed from said initiating computer to said target computer as XML data." Note should be taken that such language was erroneously argued with respect to the independent claims in the pre-appeal remarks mailed June 12, 2006. However, it should be strongly noted that the independent claims are not currently limited by such language.

Appellant respectfully asserts that such excerpt only teaches an mBot located on a client side that implements XML. Simply nowhere in such excerpt is there even a suggestion of any sort of "operation specifying data," let alone "operation specifying data [that] is passed from said initiating computer to said target computer as XML data" (emphasis added), as claimed by appellant.

- 18 -

Again, appellant respectfully asserts that at least the third element of the *prima facie* case of obviousness has not been met, since the prior art references, when combined, fail to teach or suggest all of the claim limitations, as noted above.

Group #3: Claims 3, 20, and 37

With respect to Claims 3, 20, and 37, the Examiner has relied on paragraph [0065] lines 1-14 in Uszok to make a prior art showing of appellant's claimed technique "wherein said operation specifying data represents said target process as a complex data type within said XML data."

Appellant respectfully asserts that such excerpt only teaches that "[t]he user's actions are captured as data messages and routed by the botMaster core 530 to the Knowledge Manager component 590 for recording in the User Model 592" (emphasis added). Further, Uszok teaches that the user model "may be imported from external sources, for example using RDF" (emphasis added), which "can use XML as an interchange syntax." However, the mere disclosure of capturing User Model data and importing User Model data using RDF, which can use XML as an interchange syntax, as in Uszok, does not even suggest the technique "wherein said operation specifying data represents said target process as a complex data type within said XML data" (emphasis added), as specifically claimed by appellant.

Again, appellant respectfully asserts that at least the third element of the *prima facie* case of obviousness has not been met, since the prior art references, when combined, fail to teach or suggest all of the claim limitations, as noted above.

Group #4: Claims 4, 21, and 38

With respect to Claims 4, 21, and 38, the Examiner has relied on paragraphs [0069], [0077], and [0088] in Uszok to make a prior art showing of appellant's claimed technique "wherein parameter data used by said target process is represented by data within said complex data type of said target process."

- 19 -

Appellant respectfully asserts that the excerpts cited by the Examiner merely teach the use of a "suitable botServer to provide botBox hosting [in order for] the user's botMaster [to] communicate with the selected botServer." The cited excerpts also teach that the botServer provides "a forum for sBots to meet and exchange information" and "secure access to proprietary information based on XML schemas." In addition, the excerpts teach that "the botServer is controlled primarily by a botServer manager (core) component." Furthermore, the cited references teach the use of SOAP (Simple Object Access Protocol) in order "for a program running in one kind of operating system... to communicate with a program in the same or another kind of an operating system... by using [HTTP and XML] as the mechanisms for information exchange." However, the mere disclosure that the running program communicates with another program via HTTP and XML simply fails to even suggest a technique "wherein parameter data used by said target process is represented by data within said complex data type of said target process" (emphasis added), as claimed by appellant.

Again, appellant respectfully asserts that at least the third element of the *prima facie* case of obviousness has not been met, since the prior art references, when combined, fail to teach or suggest all of the claim limitations, as noted above.

> *Group #5: Claims 9, 26, and 43*

With respect to Claims 9, 26, and 43, the Examiner has relied on paragraph [0070] lines 18-20 in Uszok to make a prior art showing of appellant's claimed technique "wherein said operation specifying data includes parameter data used by said target process in said operation."

Appellant respectfully asserts that the excerpt relied upon by the Examiner merely discloses that "the botMaster communicator component 408 then sends an appropriate message back to the requesting botMaster, including identification of the newly created botBox account, preferably, a guid [globally unique identifier]." Clearly, the message described by Uszok only identifies a newly created botBox, which clearly fails to even suggest any sort of operation specifying data, let alone where such "operation specifying data includes parameter data used by said target process in said operation" (emphasis added), as claimed by appellant.

- 20 -

Again, appellant respectfully asserts that at least the third element of the *prima facie* case of obviousness has not been met, since the prior art references, when combined, fail to teach or suggest <u>all</u> of the claim limitations, as noted above.

Group #6: Claims 14, 31, and 48

With respect to Claims 14, 31, and 48, the Examiner has relied on paragraphs [0088] and [0128] in Uszok to make a prior art showing of appellant's claimed technique "wherein said result data is passed from said target computer to said initiating computer as XML data."

Appellant respectfully asserts that the excerpts cited by the Examiner disclose, in part, the use of SOAP (Simple Object Access Protocol) in order "for a program running in one kind of operating system... to communicate with a program in the same or another kind of an operating system... by using [HTTP and XML] as the mechanisms for information exchange." In addition, Uszok discloses that "[i]nteraction protocols define a set of states that a bot may exist in, rules of transition from one state to another and [optionally] a description of the structure (schema) of documents (data) that can be exchanged between participating parties in order to transition from one state to another" where the "interaction protocol can be described... in an XML-based language." Clearly, the mere disclosure that data may be <u>exchanged</u> between participating parties using an XML-based language simply fails to suggest <u>result data</u>, much less a technique "wherein said <u>result data is passed</u> **from said target computer to said initiating computer as XML data**" (emphasis added), as claimed by appellant.

Again, appellant respectfully asserts that at least the third element of the *prima facie* case of obviousness has not been met, since the prior art references, when combined, fail to teach or suggest <u>all</u> of the claim limitations, as noted above.

Group #7: Claims 15, 32, and 49

With respect to Claims 15, 32, and 49, the Examiner has relied on paragraphs [0050] and [0068] in Uszok to make a prior art showing of appellant's claimed technique "wherein said operation

includes returning said result data from said target computer to said initiating computer in dependence upon whether or not said target process is available to said target computer."

Appellant respectfully asserts that such excerpts only generally disclose bots ([0050]) and a method for installing a botBox ([0068]), but do not disclose a situation where result data is returned in dependence on the availability of a bot. Appellant emphasizes that nowhere in such excerpts, or the entire Uszok reference, is there any disclosure of "returning said result data... in dependence upon whether or not said target process is available to said target computer" (emphasis added), as specifically claimed by appellant.

Again, appellant respectfully asserts that at least the third element of the *prima facie* case of obviousness has not been met, since the prior art references, when combined, fail to teach or suggest all of the claim limitations, as noted above.

Group #8: Claims 17, 34, and 51

With respect to Claims 17, 34, and 51, the Examiner has relied on paragraph [0073] lines 14-21 in Uszok to make a prior art showing of appellant's claimed technique "wherein said operation specifying data is validated by said target computer by comparing with a template defining valid data."

Appellant respectfully asserts that such excerpt only teaches "checking digital signatures" and that "the Bot manifest is verified against the description file to validate the privileges that are to be granted." Clearly, a digital signature does not meet appellant's claimed "comparing [said operating specifying data] with a template defining valid data." Furthermore, Uszok's description file that is merely used to validate privileges also does not meet "operation specifying data [that] is validated by said target computer," as claimed by appellant.

Again, appellant respectfully asserts that at least the third element of the *prima facie* case of obviousness has not been met, since the prior art references, when combined, fail to teach or suggest all of the claim limitations, as noted above.

- 22 -

Group #9: Claim 52

With respect to Claim 52, the Examiner has relied on Col. 7, lines 24-32 in Kouznetsov to make a prior art showing of appellant's claimed "validating said operation specifying data received at said agent process against schema data, where said schema data is sent to said agent process from said initiating computer at the same time as said operation specifying data."

Appellant respectfully asserts that such excerpt only discloses that code must include a <u>digital signature</u>. Clearly, a digital signature does not meet appellant's claimed "schema data [that] is sent to said agent process from said initiating computer <u>at the same time as said operation specifying data</u>", and especially not where "operation specifying data received at said agent process [is validated] against [the] schema data," (emphasis added), in the context claimed.

Again, appellant respectfully asserts that at least the third element of the *prima facie* case of obviousness has not been met, since the prior art references, when combined, fail to teach or suggest <u>all</u> of the claim limitations, as noted above.

Group #10: Claim 54

With respect to Claim 54, the Examiner has relied on Col. 7, lines 27-32 in Kouznetsov to make a prior art showing of appellant's claimed "parsing said operation specifying data after validating said operation specifying data to extract at least one identifier for mapping said at least one identifier to an available target process."

Appellant respectfully asserts that such excerpt only generally discloses security functions such as methods for performing authentication and validation functions. Simply nowhere does Kouznetsov even suggest any sort of parsing, let alone "parsing said operation specifying data after validating said operation specifying data to extract at least one identifier for mapping said at least one identifier to an available target process" (emphasis added), as claimed by appellant.

- 23 -

Again, appellant respectfully asserts that at least the third element of the *prima facie* case of obviousness has not been met, since the prior art references, when combined, fail to teach or suggest all of the claim limitations, as noted above.

In view of the remarks set forth hereinabove, all of the independent claims are deemed allowable, along with any claims depending therefrom.

- 24 -

VIII CLAIMS APPENDIX (37 C.F.R. § 41.37(c)(1)(viii))

The text of the claims involved in the appeal (along with associated status information) is set forth below:

1.   (Previously Presented) A computer program product for controlling a target computer to perform an operation in response to data received from an initiating computer, said computer program product comprising:

agent process code operable to execute on said target computer to provide an agent process to:

receive at an agent process executing on said target computer autonomously generated operation specifying data sent from said initiating computer to said target computer;

read from said operation specifying data an identifier of a target process for performing said operation; and

if said target process is available to said target computer to pass at least a portion of said operation specifying data from said agent process to said target process; and

target process code operable to provide one or more target processes for performing operations in response to operation specifying data, said one or more target processes being provided at said target computer independently of said agent process;

wherein said operation performed includes configuring said target computer to execute a computer program;

wherein said target process is operable to map configuration data specified within said operation specifying data to a configuration data store of said target computer;

wherein said configuration data store is one of:

a Windows Registry entry;

an INI file;

a DAPI store; and

a database entry;

wherein said identifier of a target process includes at least one of:

data specifying a computer file operable to trigger said target process;

- 25 -

data specifying a communication channel operable to trigger said target process; and

data specifying an operating system command operable to trigger said target process;

wherein said operation includes returning result data from said target computer to said initiating computer in dependence upon said operation performed by said target process;

wherein said result data includes data specifying existing configuration data of said target computer;

wherein said target process is operable to map said existing configuration data of said target computer stored within said configuration data store of said target computer to said result data to be returned to said initiating computer.

2.   (Original) A computer program product as claimed in claim 1, wherein said operation specifying data is passed from said initiating computer to said target computer as XML data.

3.   (Previously Presented) A computer program product as claimed in claim 2, wherein said operation specifying data represents said target process as a complex data type within said XML data.

4.   (Previously Presented) A computer program product as claimed in claim 3, wherein parameter data used by said target process is represented by data within said complex data type of said target process.

5.   (Cancelled)

6.   (Cancelled)

7.   (Cancelled)

8.   (Cancelled)

- 26 -

9. (Original) A computer program product as claimed in claim 1, wherein said operation specifying data includes parameter data used by said target process in said operation.

10. (Cancelled)

11. (Cancelled)

12. (Cancelled)

13. (Cancelled)

14. (Previously Presented) A computer program product as claimed in claim 1, wherein said result data is passed from said target computer to said initiating computer as XML data.

15. (Previously Presented) A computer program product as claimed in claim 1, wherein said operation includes returning said result data from said target computer to said initiating computer in dependence upon whether or not said target process is available to said target computer.

16. (Original) A computer program product as claimed in claim 1, wherein an operation that may be performed by said target computer includes installing a new target process.

17. (Original) A computer program product as claimed in claim 1, wherein said operation specifying data is validated by said target computer by comparing with a template defining valid data.

- 27 -

18.   (Previously Presented) A method of controlling a target computer to perform an operation in response to data received from an initiating computer, said method comprising the steps of:

receiving at an agent process executing on said target computer autonomously generated operation specifying data sent from said initiating computer to said target computer;

reading from said operation specifying data an identifier of a target process for performing said operation;

if said target process is available to said target computer, then passing at least a portion of said operation specifying data from said agent process to said target process; and

performing said operation using said target process; wherein

one or more target processes for performing operations in response to operation specifying data are provided at said target computer independently of said agent process;

wherein said operation performed includes configuring said target computer to execute a computer program;

wherein said target process is operable to map configuration data specified within said operation specifying data to a configuration data store of said target computer;

wherein said configuration data store is one of:

a Windows Registry entry;

an INI file;

a DAPI store; and

a database entry;

wherein said identifier of a target process includes at least one of:

data specifying a computer file operable to trigger said target process;

data specifying a communication channel operable to trigger said target process; and

data specifying an operating system command operable to trigger said target process;

wherein said operation includes returning result data from said target computer to said initiating computer in dependence upon said operation performed by said target process;

- 28 -

wherein said result data includes data specifying existing configuration data of said target computer;

wherein said target process is operable to map said existing configuration data of said target computer stored within said configuration data store of said target computer to said result data to be returned to said initiating computer.

19.   (Original) A method as claimed in claim 18, wherein said operation specifying data is passed from said initiating computer to said target computer as XML data.

20.   (Previously Presented) A method as claimed in claim 19, wherein said operation specifying data represents said target process as a complex data type within said XML data.

21.   (Previously Presented) A method as claimed in claim 20, wherein parameter data used by said target process is represented by data within said complex data type of said target process.

22.   (Cancelled)

23.   (Cancelled)

24.   (Cancelled)

25.   (Cancelled)

26.   (Original) A method as claimed in claim 18, wherein said operation specifying data includes parameter data used by said target process in said operation.

27.   (Cancelled)

28.   (Cancelled)

- 29 -

29.   (Cancelled)

30.   (Cancelled)

31.   (Previously Presented) A method as claimed in claim 18, wherein said result data is passed from said target computer to said initiating computer as XML data.

32.   (Previously Presented) A method as claimed in claim 18, wherein said operation includes returning said result data from said target computer to said initiating computer in dependence upon whether or not said target process is available to said target computer.

33.   (Original) A method as claimed in claim 18, wherein an operation that may be performed by said target computer includes installing a new target process.

34.   (Original) A method as claimed in claim 18, wherein said operation specifying data is validated by said target computer by comparing with a template defining valid data.

35.   (Previously Presented) Apparatus for controlling a target computer to perform an operation in response to data received from an initiating computer, said apparatus comprising:
    agent process logic of said target computer to provide an agent process to:
    receive at an agent process executing on said target computer autonomously generated operation specifying data sent from said initiating computer to said target computer;
    read from said operation specifying data an identifier of a target process for performing said operation; and
    if said target process is available to said target computer to pass at least a portion of said operation specifying data from said agent process to said target process; and

- 30 -

target process logic operable to provide one or more target processes for performing operations in response to operation specifying data, said one or more target processes being provided at said target computer independently of said agent process;

wherein said operation performed includes configuring said target computer to execute a computer program;

wherein said target process is operable to map configuration data specified within said operation specifying data to a configuration data store of said target computer;

wherein said configuration data store is one of:

a Windows Registry entry;

an INI file;

a DAPI store; and

a database entry;

wherein said identifier of a target process includes at least one of:

data specifying a computer file operable to trigger said target process;

data specifying a communication channel operable to trigger said target process; and

data specifying an operating system command operable to trigger said target process;

wherein said operation includes returning result data from said target computer to said initiating computer in dependence upon said operation performed by said target process;

wherein said result data includes data specifying existing configuration data of said target computer;

wherein said target process is operable to map said existing configuration data of said target computer stored within said configuration data store of said target computer to said result data to be returned to said initiating computer.

36.   (Original) Apparatus as claimed in claim 35, wherein said operation specifying data is passed from said initiating computer to said target computer as XML data.

- 31 -

37.   (Previously Presented) Apparatus as claimed in claim 36, wherein said operation specifying data represents said target process as a complex data type within said XML data.

38.   (Previously Presented) Apparatus as claimed in claim 37, wherein parameter data used by said target process is represented by data within said complex data type of said target process.

39.   (Cancelled)

40.   (Cancelled)

41.   (Cancelled)

42.   (Cancelled)

43.   (Original) Apparatus as claimed in claim 35, wherein said operation specifying data includes parameter data used by said target process in said operation.

44.   (Cancelled)

45.   (Cancelled)

46.   (Cancelled)

47.   (Cancelled)

48.   (Previously Presented) Apparatus as claimed in claim 35, wherein said result data is passed from said target computer to said initiating computer as XML data.

- 32 -

49.   (Previously Presented) Apparatus as claimed in claim 35, wherein said operation includes returning said result data from said target computer to said initiating computer in dependence upon whether or not said target process is available to said target computer.

50.   (Original) Apparatus as claimed in claim 35, wherein an operation that may be performed by said target computer includes installing a new target process.

51.   (Original) Apparatus as claimed in claim 35, wherein said operation specifying data is validated by said target computer by comparing with a template defining valid data.

52.   (Previously Presented) A computer program product as claimed in claim 1, further comprising validating said operation specifying data received at said agent process against schema data, where said schema data is sent to said agent process from said initiating computer at the same time as said operation specifying data.

53.   (Previously Presented) A computer program product as claimed in claim 1, further comprising validating said operation specifying data received at said agent process against schema data, where said schema data is present in said agent process when said operation specifying data is sent.

54.   (Previously Presented) A computer program product as claimed in claim 1, further comprising parsing said operation specifying data after validating said operation specifying data to extract at least one identifier for mapping said at least one identifier to an available target process.

IX EVIDENCE APPENDIX (37 C.F.R. § 41.37(c)(1)(ix))

There is no such evidence.

- 34 -

## X RELATED PROCEEDING APPENDIX (37 C.F.R. § 41.37(c)(1)(x))

There is no such related proceeding.

- 35 -

In the event a telephone conversation would expedite the prosecution of this application, the Examiner may reach the undersigned at (408) 971-2573. For payment of any additional fees due in connection with the filing of this paper, the Commissioner is authorized to charge such fees to Deposit Account No. 50-1351 (Order No. NAI1P451/01.285.01).

Respectfully submitted,
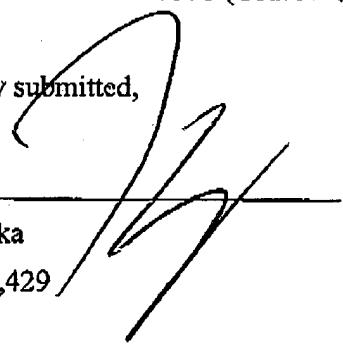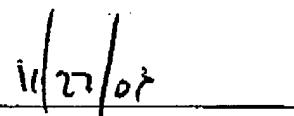
By: _____   Date: ___11/27/06_____
Kevin J. Zilka
Reg. No. 41,429

Zilka-Kotab, P.C.
P.O. Box 721120
San Jose, California 95172-1120
Telephone: (408) 971-2573
Facsimile: (408) 971-4660